

# The effectiveness of position- and composition-specific gap costs for protein similarity searches

Aleksandar Stojmirović, E. Michael Gertz, Stephen F. Altschul, and Yi-Kuo Yu\*

National Center for Biotechnology Information  
National Library of Medicine  
National Institutes of Health  
Bethesda, MD 20894  
United States

## Abstract

### Motivation:

The flexibility in gap cost enjoyed by Hidden Markov Models (HMMs) is expected to afford them better retrieval accuracy than position-specific scoring matrices (PSSMs). We attempt to quantify the effect of more general gap parameters by separately examining the influence of position- and composition-specific gap scores, as well as by comparing the retrieval accuracy of the PSSMs constructed using an iterative procedure to that of the HMMs provided by Pfam and SUPERFAMILY, curated ensembles of multiple alignments.

### Results:

We found that position-specific gap penalties have an advantage over uniform gap costs. We did not explore optimizing distinct uniform gap costs for each query. For Pfam, PSSMs iteratively constructed from seeds based on HMM consensus sequences perform equivalently to HMMs that were adjusted to have constant gap transition probabilities, albeit with much greater variance. We observed no effect of composition-specific gap costs on retrieval performance.

### Availability:

The scripts for performing evaluations are available upon request from the authors.

### Contact:

yyu@ncbi.nlm.nih.gov

---

\*to whom correspondence should be addressed

# 1 Introduction

Information retrieval from molecular databases by sequence alignment is an essential component of modern biology. The effectiveness of retrieval strategies depends crucially on how alignments are scored. A pairwise alignment score typically combines scores for the substitutions, insertions, and deletions that transform one sequence into another. Scores for substitutions are derived from a substitution matrix, while scores for insertions and deletions are known as gap costs. The importance of gap costs has prompted numerous studies proposing various reasonable gap penalty schemes [25, 4, 28, 6, 35, 27].

Search accuracy may be improved substantially by using position-specific scoring matrices (PSSM) [15]. In addition, it is possible to introduce position- and composition-specific gap costs, which so far have been implemented primarily by Hidden Markov Models (HMMs) [22, 7]. In this paper we attempt to quantify the effect of different gap scores on retrieval performance using PSI-BLAST [1] and HMMER [8, 9], canonical examples of software tools employing PSSMs and HMMs, respectively.

As its name suggests, a PSSM assigns scores to amino acids in a database sequence based on the position in which they occur in the alignment. PSI-BLAST computes and scores alignments using a heuristic approximation to the Smith-Waterman algorithm [30] with affine gap costs [11] providing uniform penalties for opening and extending a gap. PSSMs used by PSI-BLAST may be generated through an iterative search procedure, or obtained from other sources, such as databases of curated multiple sequence alignments (MSAs).

Two publicly available sources of curated alignments are the Pfam [10] and SUPERFAMILY [12, 33] databases. In both, each MSA is represented by an HMM, which may be used for similarity searches. An HMM is a finite-state automaton, characterized by state-to-state transition probabilities and emission probabilities that generate hypothetical protein sequences. See Fig. 1 for an example and the Appendix for more details.

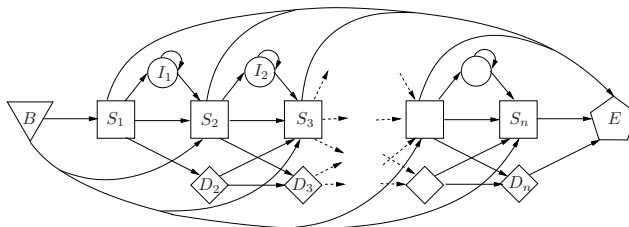


Figure 1: An example of a protein profile HMM architecture used by HMMER. The model contains  $n$  positions plus a begin state ( $B$ ) and end state ( $E$ ). Each position contains a substitution ( $S$ ) and a deletion state ( $D$ ), with a possible insertion state ( $I$ ) between two  $S$ -nodes. Allowed transitions are shown by arrows. To simulate local alignments, transitions  $B \rightarrow S_i$  and  $S_i \rightarrow E$ , for any  $S_i$ , are permitted.

The HMMER package [8, 9] uses the Viterbi algorithm [7], which finds the highest-scoring sequence of states in the HMM that produces the database sequence. The probability that a particular amino acid is emitted in a HMMER substitution state may be identified with the probability that it occurs in a corresponding position in a PSI-BLAST PSSM. On the other hand, HMMER allows position- and composition-specific gap scores, which model the probability that an insertion or deletion occurs at a particular position in an

alignment.

With their greater gap cost flexibility, HMMs may be expected to have better retrieval accuracy than PSSMs. We attempt to quantify the effect of HMMER’s use of more general gap parameters by separately examining the influence of position- and composition-specific gap scores. We also compare the retrieval accuracy of the PSSMs constructed using PSI-BLAST’s iterative procedure to that of the HMMs provided by the Pfam and SUPERFAMILY collections. Our results may suggest some directions for improvements to PSI-BLAST, and the magnitude of the improvements one might expect.

We collected protein profile HMMs from Pfam and SUPERFAMILY. We then modified the profiles from each source to simulate different retrieval strategies, and used them as queries for HMMER and PSI-BLAST to search a set of sequences from the SCOP (structural classification of proteins) database [24, 2], which forms our ‘gold standard’. We use the results of the searches to evaluate and compare the retrieval performance of the search methods considered.

SCOP is a database of protein domains, classified by structure, function and sequence. Protein domains are classified into a hierarchy of class, fold, superfamily and family. Domains sharing the same superfamily are assumed to be homologous. For our testing purposes, we use the ASTRAL 40 [5] subset of SCOP (release 1.71), consisting of domain sequences that were filtered so that no two sequences share more than 40% pairwise identity. ASTRAL has been used as the testing set in a number of performance evaluations of protein sequence comparison algorithms [13, 31, 26, 38].

It is generally useful to evaluate not only the difference in performance of two search methods, but also whether such a difference is statistically significant. A number of procedures have been proposed, mostly based on bootstrap resampling with replacement [13, 26]. In this context, Green and Brenner [13] observed that large superfamilies have an undue influence on the results, as the number of possible relationships grows quadratically with the number of members in a superfamily. They therefore proposed two weighting schemes that reduce the influence of large superfamilies. Price *et al.* [26] noted technical challenges in obtaining accurate variances for the weighted statistics and proposed an improved bootstrap.

Our query sets, based on Pfam and SUPERFAMILY, contain several models for each SCOP-classified superfamily. Some superfamilies are overrepresented both in the query sets and in the ASTRAL database. We propose a different method than Price *et al.* [26] to address the difficulties associated with having superfamilies of different sizes. Our strategy is to sample without replacement 3/4 of the superfamilies and then select a single model for each superfamily in any given query set. Hence, each sample contains no more than a single profile from each superfamily and therefore captures the most distant relationships among queries.

## 2 Materials and Methods

### Software tools

For HMM-based queries, we used the HMMER package (version 2.3.2) [8, 9], which is also used internally by Pfam. Local alignment between a sequence and an HMM is allowed by the nonzero probabilities of entering match nodes directly from the begin state, as well as moving directly to the end state from them (Fig. 1). The statistical significance of each alignment score is estimated using an assumed extreme value distribution, with model-specific parameters. The final E-value, adjusted for model and sequence composition, is used to rank the hits. Another popular HMM platform is SAM [20, 3, 21], which is used by SUPERFAMILY.

We used HMMER rather than SAM for all our HMM-based queries because the programs’ retrieval performances were shown to be comparable [23, 34] and because the SUPERFAMILY models were available in HMMER format.

For PSSM-based queries, we used PSI-BLAST (version 2.2.17) [1]. The statistics of PSI-BLAST scores are based on the extreme value distribution [16] with a correction for finite sequence length. The statistical significance of each database hit is refined by taking into account its composition as well as that of the PSSM [29].

PSI-BLAST allows one to start a search from a ‘checkpoint’ file containing a PSSM saved from an earlier PSI-BLAST run, or built by other means. In addition to a PSSM, PSI-BLAST requires gap penalties as input: a gap opening cost and a gap extension cost. The choice of gap penalties is restricted to a few values because the parameters required to produce accurate statistics are precomputed using large-scale simulations. For both HMMER and PSI-BLAST runs, we used the standard search executables with their default settings.

## Query sets

Following Wistrand and Sonnhammer [34], we constructed a query set of Pfam (release 22.0) models by identifying all Pfam-A models that were cross referenced by Pfam with an identifier in SCOP 1.71, and mapping the cross-referenced SCOP identifier to a SCOP superfamily. We did not consider models that have multiple domains mapping to different superfamilies.

We filtered the resulting set of Pfam models using two additional rules. First, any model mapping to a SCOP superfamily that had fewer than four members in ASTRAL 40 was removed from further consideration, to avoid superfamilies with a small number of members from disproportionately influencing the results. Next, we examined the MSA used to generate the Pfam profile and kept only those families whose MSA contained at least 10 sequences and had an average sequence length of at least 30 amino acids. Our final Pfam query set contained 703 Pfam models representing 299 superfamilies. We used the profiles from the `Pfam_fs` set, built for local/local alignment.

Our second query set consisted of all 6729 models from the SUPERFAMILY database (release 1.69) that belonged to the 299 superfamilies in the Pfam query set. These models were also built for local/local alignment. The above query sets, paired with HMMER, formed our first two search methods, which we named HOF (HMM, ‘original’, Pfam) and HOU (HMM, ‘original’, SUPERFAMILY).

The second pair of search methods, called HBF and HBU (see Table 1 for an outline of all search methods), was constructed by taking the HMMs from HOF and HOU, respectively, and replacing all emission scores for each insert state with 0. This is equivalent to setting all insertion emission probabilities to the background probabilities.

We constructed the third pair of search methods, called HGF and HGU by taking the HMMs from HBF and HBU, respectively, and adjusting the state transition probabilities to correspond to those implied by the affine gap penalties used by PSI-BLAST (see Appendix for a detailed explanation).

Let  $\alpha$  denote the gap opening cost and  $\beta$  the gap extension cost, in bits. We used the default penalty of PSI-BLAST, which is 11 ( $\alpha = 5.040$  bits) for gap opening and 1 for gap extension ( $\beta = 0.458$  bits). This scale was chosen to match the scale of BLOSUM62 [19], the default scoring matrix of BLAST.

For each position  $m$  of an HMM, we left the probabilities  $P(B \rightarrow S_m)$  and  $e_m = P(S_m \rightarrow E)$  un-

Name	Description
HO	Original HMM dataset.
HB	HMMs, background insertion emission probabilities
HG	HMMs, constant state transitions and background insertion emissions
PO	PSSMs, converted from original HMMs.
PC	PSSMs, from 5 PSI-BLAST iterations over nr using profile consensus seeds
PS	PSSMs, from 5 PSI-BLAST iterations over nr using SCOP domain sequence seeds

Table 1: Nomenclature of query sets. As shown in this table, the first two letters of the abbreviations of various search strategies denote the type of profile (HMM or PSSM), and the method of construction. The third letter is optionally appended to show the database of origin (F for Pfam, U for SUPERFAMILY).

changed and set the remaining transition probabilities as follows:

$$P(D_m \rightarrow D_{m+1}) = P(I_m \rightarrow I_m) = \nu, \quad (1)$$

$$P(D_m \rightarrow S_{m+1}) = P(I_m \rightarrow S_{m+1}) = 1 - \nu, \quad (2)$$

$$P(S_m \rightarrow D_{m+1}) = P(S_m \rightarrow I_m) = \frac{\mu(1 - e_m)}{1 + 2\mu - \nu}, \quad (3)$$

$$P(S_m \rightarrow S_{m+1}) = \frac{(1 - e_m)(1 - \nu)}{1 + 2\mu - \nu}, \quad (4)$$

where  $\mu = 2^{\alpha+\beta}$  and  $\nu = 2^\beta$ . The probabilities were read from HMMER files, converted from scores, modified and written back as scores, as per HMMER convention [9]. After modification, the HMMER statistical parameters of each HMM of HBF, HBU, HGF and HGU were recalibrated.

The remaining search methods used PSI-BLAST with default gap penalties. POF and POU used PSSMs derived from HOF and HOU, respectively, by taking the match state emission probabilities and writing them in PSI-BLAST checkpoint format. PCF and PCU used PSSMs obtained using the standard PSI-BLAST iterative procedure. We obtained the consensus (most likely) sequences of POF and POU profiles and used them as seeds for the initial searches, running 5 iterations in total against nr, the database of non-redundant protein sequences maintained by NCBI (frozen on Apr 11, 2007) [32].

The final search method, named PSU used the same construction procedure as POU except that the SCOP sequences associated with SUPERFAMILY models were used as PSI-BLAST seeds instead of profile consensus sequences.

## Performance evaluation

As described above, our query sets contained no profiles assigned to more than one SCOP superfamily. Each pair  $\mathbf{p}, \mathbf{s}$ , where  $\mathbf{p}$  is a query profile and  $\mathbf{s}$  is an ASTRAL sequence, was classified as similar (‘positive’) if  $\mathbf{s}$  belongs to the superfamily associated with  $\mathbf{p}$ , and not similar (‘negative’) otherwise. For every query

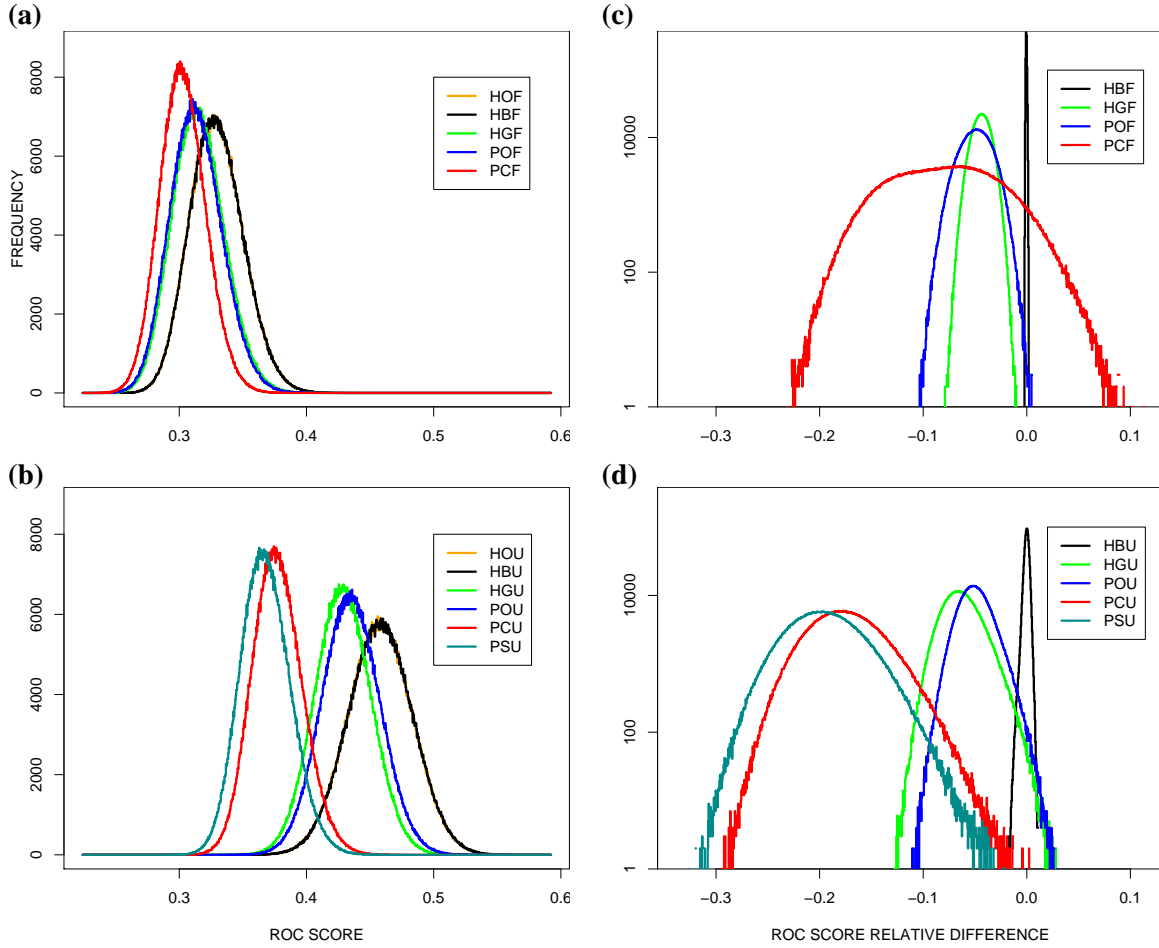


Figure 2: ROC score statistics of one million samples. In each sample, 224 superfamilies are first randomly chosen from 299 superfamilies. A representative query profile is then randomly selected from each chosen superfamily. ROC score histograms from using Pfam HMMs (a) and SUPERFAMILY HMMs (b) show appreciable difference in average ROC scores for each search method tested: SUPERFAMILY HMMs always perform better. Note that in panels (a) and (b), the curve for HO is completely covered by that for HB. Using HOF and HOU as baselines, the values of  $RRSD_{224}$  (measurement at 1 EPQ) between various methods and the baselines are computed for each sample. The resulting histograms are shown in panels (c) and (d).

$\mathbf{p}_k$  from a set of queries, denote by  $N_p(\mathbf{p}_k)$  the number of ASTRAL 40 sequences belonging to the same superfamily as  $\mathbf{p}_k$  (i.e. the total number of positives for  $\mathbf{p}_k$ ) and let  $N_p = \sum_k N_p(\mathbf{p}_k)$ .

Comparing each query profile to the ASTRAL 40 database, we retrieved a number of sequences ranked according to their E-values. These sequences were classified as true or false positives. For a given search strategy, after merging the results for the whole set of queries, we obtain the (step) functions  $p(E)$  and

$f(E)$  giving respectively the cumulative numbers of true and false positives with E-value  $E$  or smaller. The function  $p$  can also be expressed as a function of  $f$ , the number of false positives and the graph of  $p(f)$  versus  $f$  is called the ROC (receiver operating characteristic) curve [18, 14, 17]. The same curve can be displayed as a coverage vs. error-per-query (EPQ) or CVE plot.

Our main performance statistic is the (truncated) ROC score. Given a number of false positives  $F$ , the  $\text{ROC}_F$  score is defined by

$$\text{ROC}_F = \frac{1}{F \cdot N_p} \int_0^F p(f) df. \quad (5)$$

It represents the accuracy of the search method (given a set of queries) for a given number of false positives. To compare two search methods  $M_1$  and  $M_2$  we compute their relative  $\text{ROC}_F$  score difference, denoted  $\text{RRSD}_F$ , defined by

$$\text{RRSD}_F(M_1, M_2) \equiv \frac{\text{ROC}_F(M_1) - \text{ROC}_F(M_2)}{\text{ROC}_F(M_2)}. \quad (6)$$

To overcome the aforementioned problems associated with overrepresentation of large superfamilies, we sampled according to the superfamily classification. For each sample we randomly picked 224 out of 299 superfamilies (leaving 1/4 out) without replacement. Then, we selected one representative profile for each superfamily to form a sample query set. Search methods using the profiles originating from the same source (Pfam or SUPERFAMILY) used the same samples so that their performances could be compared for each sample. Our main statistic is the  $\text{RRSD}_{224}$  per sample, which measures performance at 1 EPQ or less. It allows a fair comparison of search methods.

### 3 Results

Fig. 2 shows the distributions of  $\text{ROC}_{224}$  scores and their relative differences ( $\text{RRSD}_{224}$ ) *per sample* with respect to HO for all query sets. Comparison of Fig. 2 (a) and (b) shows that, in general, the strategies using profiles from SUPERFAMILY perform better than those using Pfam profiles. In terms of relative difference (Fig. 2 (c,d), Table 2), using both Pfam and SUPERFAMILY profiles, original HMMs (HO) perform significantly better than all other query sets except HB. There is no perceivable difference between HB and HO. There is also no significant difference between HG and PO.

In the case of PSSMs, POU gives better performance than PCU and PSU, but there is no significant difference between POF and PCF, although PCF shows a large variance in performance. In a number of cases, a PCF sample even outperforms the corresponding HOF sample. The relative ROC score difference between PCU and PSU is slightly positive, but not significantly so.

Using profiles from Pfam (SUPERFAMILY), we observed two (three) clusters of search strategies that performed equivalently based on  $\text{RRSD}_{224}$  (Fig. 2 (c,d)). This trend in performance is supported by Fig. 3, which displays examples of CVE curves for all alignment methods tested. The samples associated with these CVE curves have the median  $\text{ROC}_{224}$  score.

(a)	HOF			HBF			HGF			POF			PCF		
HOF	0.0	0.0	0.0	-0.1	0.0	0.1	2.9	4.5	6.3	2.4	5.1	8.1	-0.6	8.5	19.5
HBF	-0.1	-0.0	0.1	0.0	0.0	0.0	2.9	4.5	6.3	2.4	5.1	8.1	-0.6	8.5	19.4
HGF	-5.9	-4.3	-2.8	-5.9	-4.3	-2.8	0.0	0.0	0.0	-1.7	0.5	3.0	-4.9	3.8	14.1
POF	-7.5	-4.8	-2.3	-7.5	-4.8	-2.4	-3.0	-0.5	1.8	0.0	0.0	0.0	-5.1	3.2	13.1
PCF	-16.3	-7.8	0.6	-16.3	-7.8	0.6	-12.3	-3.7	5.1	-11.6	-3.1	5.3	0.0	0.0	0.0

(b)	HOU			HBU			HGU			POU			PCU			PSU		
HOU	0.0	0.0	0.0	-0.4	-0.0	0.4	2.9	6.9	10.2	2.0	5.3	8.2	12.4	21.4	30.3	15.0	24.5	34.2
HBU	-0.4	0.0	0.4	0.0	0.0	0.0	2.9	6.9	10.2	2.1	5.3	8.2	12.4	21.4	30.3	15.0	24.5	34.2
HGU	-9.3	-6.4	-2.8	-9.3	-6.4	-2.8	0.0	0.0	0.0	-4.1	-1.4	1.4	7.0	13.6	20.8	9.6	16.4	24.3
POU	-7.6	-5.0	-2.0	-7.6	-5.1	-2.0	-1.4	1.5	4.3	0.0	0.0	0.0	8.1	15.3	22.8	10.7	18.2	26.3
PCU	-23.3	-17.6	-11.0	-23.3	-17.7	-11.0	-17.2	-12.0	-6.6	-18.6	-13.3	-7.5	0.0	0.0	0.0	-1.5	2.5	7.2
PSU	-25.5	-19.6	-13.0	-25.5	-19.7	-13.1	-19.5	-14.1	-8.8	-20.8	-15.4	-9.7	-6.7	-2.4	1.5	0.0	0.0	0.0

Table 2: Summary of statistics of  $RRSD_{224}$  between every pair search strategies using the same source. In Fig. 2 (c) and (d), HOF and HOU were used as the baselines for Pfam and SUPERFAMILY search strategies, respectively, and the histograms of  $RRSD_{224}$  relative to the baselines are shown. It is impractical to show such histograms for all possible baselines. However, for each pair of search strategies, we may sort (in ascending order) their one million values of  $RRSD_{224}$  and record the corresponding  $RRSD_{224}$  value at various designated percentiles. In the table, there are three numbers in a row for any given pair of search strategies. As an example, the numbers 2.9, 4.5 and 6.3, associated with  $M_1 = \text{HBF}$  and  $M_2 = \text{HGF}$ , are located in the row labelled by HBF and within the column headed by HGF. Those numbers, when divided by 100, have the following interpretation: the leftmost corresponds to the  $RRSD_{224}$  value at the 2.5-th percentile, the middle to the median and the rightmost to the 97.5-th percentile. Subtable (a) records the numbers associated with Pfam search methods, while subtable (b) documents those associated with the SUPERFAMILY strategies tested.

## 4 Discussion and Conclusion

The clear separation in retrieval performance between the SUPERFAMILY and Pfam profiles could be explained by the fact that the former are based on ASTRAL sequences, which form our testing set as well. In contrast, Pfam models are based on a variety of sequence sources and were not trained on ASTRAL. Hence, a degree of overfitting the SUPERFAMILY models to the testing set, as well as the fact that ASTRAL is structure based, may explain the overall differences in performance.

Another interesting observation is that CVE curves (Fig. 3) cross at low EPQ and form distinct clusters above 0.5 EPQ. Due to small sample size, the coverage at low EPQ is expected to have a larger uncertainty, thus the crossing of CVE curves there is anticipated. At moderate EPQ, the distinct clusters indicate that the relative retrieval efficiency is not influenced by the choice of EPQ.

On both testing collections, we have observed almost no difference in performance between the original HMMs (HO) and the models derived from them having insertion emission probabilities reset to the background (HB). Examining the models in HMMER format, we found that the insertion emission distributions were almost constant over all the positions, with the common distribution being slightly biased in favor of hydrophilic amino acids. The average relative entropy between this distribution and the background distribution is very small (0.037 bits for Pfam, 0.005 bits for SUPERFAMILY), explaining the very small effect of the insertion emissions on the retrieval performance. Note that SUPERFAMILY models had higher overall probabilities of entering a gap state and hence showed a larger influence of insertion emissions than Pfam models (Fig. 2 (c,d)).



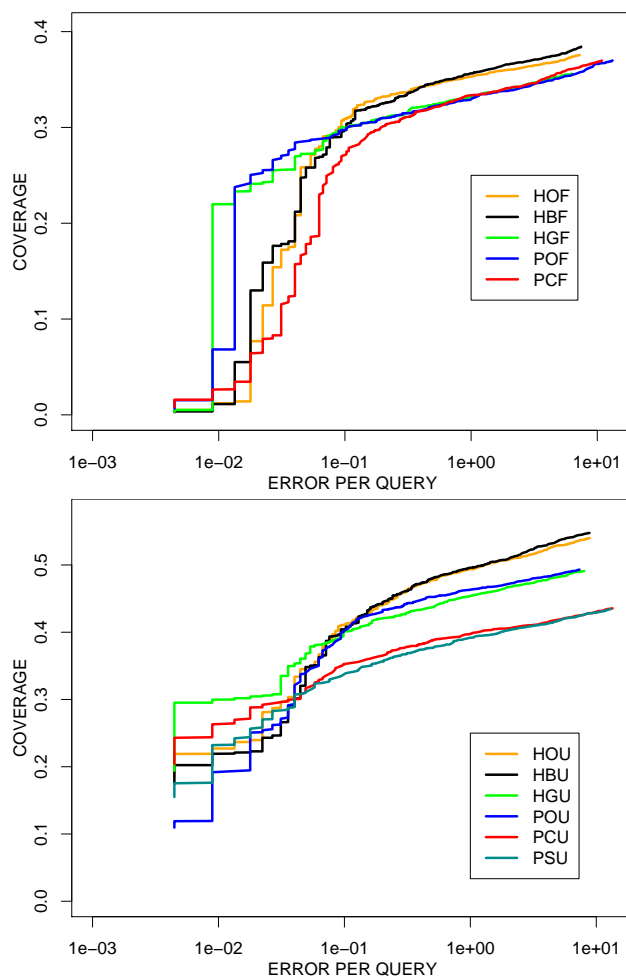


Figure 3: Example CVE curves for various search strategies based on Pfam (top) and SUPERFAMILY (bottom) profiles. Each curve shown is a representative that corresponds to a sample with  $ROC_{224}$  score equal to the median of 1,000,000 samples.

In addition, an insertion emission distribution biased in favor of hydrophilic amino acids may not be appropriate for all positions within proteins: it implicitly assumes the globular protein structure, with hydrophobic core and hydrophilic surface. Finally, from an information theoretic point of view, it is very difficult to reliably estimate insertion emission probabilities. In particular, if one wishes to establish an emission model whose emission probabilities are similar to those of the background and wants to confidently distinguish those two sets of probabilities, it is necessary to have a large amount of data. The following example illustrates this point.

In the Pfam insertion emission model, Leucine's emission probability, 0.0676, has the largest deviation compared to the background 0.0934. Consider a simple coin tossing experiment where the probability of

seeing a leucine (head) is  $p = 0.0676$  and the probability of seeing any other amino acid (tail) is  $1 - 0.0676$ . One may ask how many tosses (number of amino acids present in a gap column of an MSA) are needed in order to confidently rule out the possibility that the probability is 0.0934. It is well known that a binomial distribution in the large number limit becomes a Gaussian. In our example, the probability of observing  $k$  heads out of  $n$  tosses becomes

$$C_k^n p^k (1-p)^{n-k} \Delta k \approx \sqrt{\frac{n/p(1-p)}{2\pi}} e^{-n(\frac{k}{n}-p)^2/2p(1-p)} d(\frac{k}{n}) .$$

To reject with 85% confidence the value of 0.0934 as the probability of seeing a head, the absolute difference between the two probabilities, 0.0934 and 0.0676, must be greater than or equal to 1.02 times the standard deviation,  $\sqrt{p(1-p)/n}$ . This leads to

$$0.0258 > 1.02 \sqrt{\frac{0.0847}{n}} \Rightarrow n > 132 .$$

When applied to estimating insertion emission probabilities, this example implies one needs to have about 132 amino acids in a gap column of a multiple alignment. This number seems large for columns associated with an insert state, as these columns normally have more gaps than amino acids. On the other hand, we can confidently determine emission probabilities for columns that contain mostly amino acids and are therefore usually assigned to substitution states. Furthermore, the dominant amino acid in a match column often has very different observed and background frequencies. For example, consider a match column with 20% leucine. The same calculation as above tells us that we need only eight or more amino acids in the match column to indicate a preference for leucine. Of course, considering the sub-dominant amino acids requires more entries in the match column.

Comparing HO to HG and PO, we see that profiles with position-dependent gap parameters have 5% better retrieval performance (as measured by the median  $\text{RRSD}_{224}$  value) than those with position-independent ones. This is an area where HMMs are clearly superior to the PSSMs with constant gap penalties, as used by PSI-BLAST. Hence, a possible direction for improvement of PSI-BLAST is to introduce position-dependent gap parameters. When interpreting this difference, one should note that we did not optimize the PSI-BLAST gap penalties, but use only the defaults. It is therefore possible that the performance of PSI-BLAST with a better set of gap opening and extension penalties would more closely match the performance of HMMs. Another possibility is to estimate and optimize gap parameters for each PSSM separately, at the time of its creation (that is, each PSSM would still carry a single, position independent, gap opening and gap extension penalty, but they would not be input beforehand but estimated from the data). The practical problem with these suggested improvements is that the statistical parameters for position-specific gap penalties cannot be quickly computed as yet, and one is therefore restricted to the costs for which the parameters have been precomputed. Another possibility is to modify PSI-BLAST to use the hybrid alignment algorithm [36, 37], which is probabilistic, naturally accepts PSSMs with position-specific gap costs, and has well-characterized, universal statistics.

It is not surprising that the performances of HG and PO show no significant difference because HG was designed to simulate the PSI-BLAST gap parameters in the HMM framework. Some differences still exist due to a fundamental difference between the underlying algorithms. First, although the score statistics for HMMER and PSI-BLAST are both based on the extreme value distribution, there are still differences

in details. Second, PSI-BLAST alignments may have longer segments of ungapped alignment because the score associated with ungapped alignment is not reduced by the probability of entering another node. Some difference can also be explained by slightly different background probabilities in each case. Finally, local alignment is achieved through different mechanisms: PSI-BLAST alignments terminate when their accumulated score is maximal, while HMMER alignments terminate only when they hit the end state. Thus, HMMER alignments may tend to be more global with respect to the profile.

The difference in performance of PSI-BLAST using PSSMs constructed in different ways shows that focusing on profile construction as well as on position-specific gaps may yield significant improvement. In particular, the performance of PSSMs converted from HMMs (PO) versus those iteratively constructed (PC and PS) shows that a more carefully constructed profile may yield better performance, with the difference being more pronounced in SUPERFAMILY than in Pfam. The fact that the PSSMs obtained iteratively from `nrr` based on SUPERFAMILY consensus seeds generally perform better than those originating from Pfam consensus seeds shows the importance of the choice of the initial seed sequence. This is further emphasised by the slightly better performance of the PSSMs based on the consensus sequence as seed (PCU) than the performance of those based on the seeds taken from ASTRAL (PSU). Hence, another possible way of improving PSI-BLAST would be to run one iteration using the normal scoring matrix and construct a profile as before, but then to rerun the search using the consensus sequence as the seed instead of proceeding into the iterative stage with the profile. In that way, a more ‘central’ seed can be obtained, which, while not corresponding exactly to any sequence present in the dataset, may yield a more accurate profile for the iterative steps. Naturally, the choice of the weighting scheme for the multiple alignment used to obtain the consensus sequence or profile as well as the associated pseudocounts will also exert a significant influence on the result.

Finally, our methodology must be understood in the context of the small size of the testing suite. This does not present a significant problem when testing different parameter sets of the same alignment algorithm but when comparing different algorithms, it is essential to eliminate bias due to superfamily size. Our approach, based on sampling 3/4 of the superfamilies without replacement, was designed with this aim in mind.

## Funding

This work was supported by the Intramural Research Program of the National Library of Medicine at National Institutes of Health.

## Acknowledgement

We thank M. Wistrand and E. Sonnhammer for useful correspondence.

## References

- [1] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic*

*Acids Res.*, **25**, 3389–3402.

- [2] Andreeva, A., Howorth, D., Chandonia, J., Brenner, S., Hubbard, T., Chothia, C., and Murzin, A. (2007). Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res.*
- [3] Barrett, C., Hughey, R., and Karplus, K. (1997). Scoring hidden Markov models. *Comput. Appl. Biosci.*, **13**, 191–199.
- [4] Benner, S., Cohen, M., and Gonnet, G. (1993). Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol.*, **229**, 1065–1082.
- [5] Chandonia, J., Hon, G., Walker, N., Lo Conte, L., Koehl, P., Levitt, M., and Brenner, S. (2004). The ASTRAL Compendium in 2004. *Nucleic Acids Res.*, **32**, D189–192.
- [6] Chang, M. and Benner, S. (2004). Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J. Mol. Biol.*, **341**, 617–631.
- [7] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis*. Cambridge University press, Cambridge, UK.
- [8] Eddy, S. (1998). Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
- [9] Eddy, S. (2003). HMMER user’s guide. <ftp://selab.janelia.org/pub/software/hmmer/CURRENT/US>
- [10] Finn, R., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S., Sonnhammer, E., and Bateman, A. (2006). Pfam: clans, web tools and services. *Nucleic Acids Res.*, **34**, D247–251.
- [11] Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.
- [12] Gough, J., Karplus, K., Hughey, R., and Chothia, C. (2001). Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J. Mol. Biol.*, **313**, 903–919.
- [13] Green, R. and Brenner, S. (2002). Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proc. IEEE*, **90**(12), 1834–1847.
- [14] Gribskov, M. and Robinson, N. (1996). Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.
- [15] Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. U.S.A.*, **84**, 4355–4358.
- [16] Gumbel, E. (1958). *Statistics of Extremes*. Columbia University Press, New York.
- [17] Hajian-Tilaki, K. and Hanley, J. (2002). Comparison of three methods for estimating the standard error of the area under the curve in ROC analysis of quantitative data. *Acad. Radiol.*, **9**, 1278–1285.

- [18] Hanley, J. and McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29–36.
- [19] Henikoff, S. and Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.*, **89**, 10915–10919.
- [20] Hughey, R. and Krogh, A. (1996). Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Appl. Biosci.*, **12**, 95–107.
- [21] Karplus, K., Karchin, R., Shackelford, G., and Hughey, R. (2005). Calibrating E-values for hidden Markov models using reverse-sequence null models. *Bioinformatics*, **21**, 4107–4115.
- [22] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.
- [23] Madera, M. and Gough, J. (2002). A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Res.*, **30**, 4321–4328.
- [24] Murzin, A., Brenner, S., Hubbard, T., and Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
- [25] Pascarella, S. and Argos, P. (1992). Analysis of insertions/deletions in protein structures. *J. Mol. Biol.*, **224**, 461–471.
- [26] Price, G., Crooks, G., Green, R., and Brenner, S. (2005). Statistical evaluation of pairwise protein sequence comparison with the Bayesian bootstrap. *Bioinformatics*, **21**, 3824–3831.
- [27] Qiu, J. and Elber, R. (2006). SSALN: an alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs. *Proteins*, **62**, 881–891.
- [28] Reese, J. and Pearson, W. (2002). Empirical determination of effective gap penalties for sequence comparison. *Bioinformatics*, **18**, 1500–1507.
- [29] Schäffer, A., Aravind, L., Madden, T., Shavirin, S., Spouge, J., Wolf, Y., Koonin, E., and Altschul, S. (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.
- [30] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.
- [31] Vinga, S., Gouveia-Oliveira, R., and Almeida, J. (2004). Comparative evaluation of word composition distances for the recognition of SCOP relationships. *Bioinformatics*, **20**, 206–215.
- [32] Wheeler, D., Barrett, T., Benson, D., Bryant, S., Canese, K., Chetvernin, V., Church, D., DiCuccio, M., Edgar, R., Federhen, S., Geer, L., Kapustin, Y., Khovayko, O., Landsman, D., Lipman, D., Madden, T., Maglott, D., Ostell, J., Miller, V., Pruitt, K., Schuler, G., Sequeira, E., Sherry, S., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusov, R., Tatusova, T., Wagner, L., and Yaschenko, E. (2007). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **35**, 5–12.

- [33] Wilson, D., Madera, M., Vogel, C., Chothia, C., and Gough, J. (2007). The SUPERFAMILY database in 2007: families and functions. *Nucleic Acids Res.*, **35**, D308–313.
- [34] Wistrand, M. and Sonnhammer, E. (2005). Improved profile HMM performance by assessment of critical algorithmic features in SAM and HMMER. *BMC Bioinformatics*, **6**, 99.
- [35] Wrabl, J. and Grishin, N. (2004). Gaps in structurally similar proteins: towards improvement of multiple sequence alignment. *Proteins*, **54**, 71–87.
- [36] Yu, Y. and Hwa, T. (2001). Statistical significance of probabilistic sequence alignment and related local hidden Markov models. *J. Comput. Biol.*, **8**, 249–282.
- [37] Yu, Y., Bundschuh, R., and Hwa, T. (2002). Hybrid alignment: high-performance with universal statistics. *Bioinformatics*, **18**, 864–872.
- [38] Yu, Y., Gertz, E., Agarwala, R., Schffler, A., and Altschul, S. (2006). Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Res.*, **34**, 5966–5973.
- [39] Yu, Y.-K., Wootton, J. C., and Altschul, S. F. (2003). The compositional adjustment of amino acid substitution matrices. *Proc. Natl. Acad. Sci. U.S.A.*, **100**(26), 15688–15693.

## Appendix

The connection between the transition probabilities of HMMs for sequence evolution and the scoring function (scoring matrices and gap parameters) used in sequence comparison is elaborated in this appendix. Since such a connection has been sketched explicitly in earlier publications on hybrid alignment [36, 37], interested readers are encouraged to look into the original literature. We present a self-contained exposition here to save the reader some effort in reading through earlier papers, and to present a minor extension needed for aligning a protein sequence to a *local* HMM with explicit termination probabilities at its nodes. Note that keeping a nonzero termination probability is how HMMER achieves local alignments. Hybrid alignments achieve a local alignment by taking the maximum of the log-odd ratios at each possible termination point, and hence do not need to deal explicitly with the termination probabilities of the HMMs.

The fundamental idea of protein sequence comparison is rooted in the amino acid score (substitution) matrix, where the  $(i, j)$ th entry

$$s_{ij} = \frac{1}{\lambda} \ln \left[ \frac{Q_{ij}}{p_i p_j} \right] \quad (7)$$

is the log-odd ratio of the joint probability  $Q_{ij}$  of amino acids  $i$  and  $j$  in the target ensemble to the product of the background probabilities,  $p_i$  and  $p_j$ , of the two amino acids. Here  $\lambda$  is just a scale and is set to unity from this point on. For a valid scoring matrix [39], one has  $p_i = \sum_j Q_{ij}$  and one may express  $Q_{ij}$  as  $Q_{ij} = p_i T(j|i) = p_j T(i|j)$ , with  $T(j|i)$  being the probability for amino acid  $i$  to mutate into amino acid  $j$ . In this case, we may also write

$$s_{ij} = \ln \left[ \frac{T(j|i)}{p_j} \right], \quad (8)$$

which may now be viewed as the log-odds ratio of a conditional emission probability to the background probability.

Extending this concept [36, 37], one may score the global relatedness (alignment) between two protein sequences,  $\mathbf{a}$  and  $\mathbf{b}$ , the same way: using the log-odds ratio of  $Q[\mathbf{a}, \mathbf{b}]$  to  $P[\mathbf{a}]P[\mathbf{b}]$  (the background probability of generating a pair of random sequences  $\mathbf{a}$  and  $\mathbf{b}$ ). In terms of global relatedness,  $Q[\mathbf{a}, \mathbf{b}]$  may be regarded again as  $P[\mathbf{a}]T[\mathbf{b}|\mathbf{a}]$  and

$$\frac{Q[\mathbf{a}, \mathbf{b}]}{P[\mathbf{a}]P[\mathbf{b}]} = \frac{T[\mathbf{b}|\mathbf{a}]}{P[\mathbf{b}]}.$$
 (9)

Here  $T[\mathbf{b}|\mathbf{a}]$  is the probability for sequence  $\mathbf{a}$  to mutate into sequence  $\mathbf{b}$ . It is not hard to convince oneself that there are many different "ways" or "paths" for sequence  $\mathbf{a}$  to mutate into sequence  $\mathbf{b}$ . In fact, it has been argued that the usual *optimal* alignment corresponds to the *most probable* evolutionary path. In this context, the gap cost is related to the transition probabilities in and out of the insertion/deletion states of the HMM.

A protein HMM consists of a number of nodes. Except at the begin node, each node  $j$  allows two possible states, substitution (S) and deletion (D). The substitution state associated with node  $j$  is characterized by the transition probability from  $a_j$  to other amino acids. The deletion state is further divided into cases depending on its preceding state. In between two nodes, one can have an insertion (I) state. The transition probabilities from a given state to all other allowed states have to sum to one. Four transitions –  $S \rightarrow S$ ,  $S \rightarrow D$ ,  $D \rightarrow S$ , and  $D \rightarrow D$  – will each advance the node index by one. Transition  $S \rightarrow I$  and  $I \rightarrow S$  combined together increase effectively the node index by 1, while the transitions  $I \rightarrow I$  and  $D \rightarrow I$  (if allowed) do not change the node index at all. In many HMMs, such as the ones used by HMMER, the transitions between  $I$  and  $D$  states are strictly forbidden and we follow this rule in here to simplify our exposition.

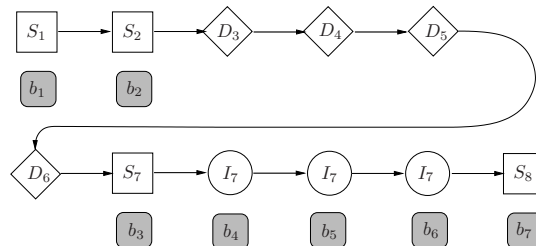


Figure 4: An example of a partial alignment between a profile HMM and a protein sequence. Note that in the text, the state preceding  $S_1$  is assumed to be a substitution state.

Constrained by the probability conservation condition, the transition probabilities are usually made node-specific (or equivalently termed *position-specific*). Focusing on the substitution scores of protein HMMs, position-specific scoring simply means that the substitution states at different nodes may emit amino acids with different sets of probabilities.

As a concrete example, let us consider an alignment of a partial HMM model of eight nodes aligned with a sequence  $\mathbf{b} = [b_1, b_2, \dots, b_7]$  of seven amino acids. Their alignment is shown in Fig. 4. The alignment

score  $S$  is given by

$$S = s_1(b_1) + s_2(b_2) + g_{(3-6)}^D + s_7(b_3) + g_{7+}^I([b_4, b_5, b_6]) + s_8(b_7) \quad (10)$$

where  $s_i(b_j)$  represents the substitution score for amino acid  $j$  at node  $i$ ,  $g_{(3-6)}^D$  represents the gap score associated with deleting nodes 3 through 6, and  $g_{7+}^I([b_4, b_5, b_6])$  represents the gap score associated with inserting amino acids  $b_4, b_5$  and  $b_6$  between nodes 7 and 8 of the HMM. The superscript "+" associated with the  $I$  state will be suppressed from this point on. The probability of occurrence associated with this alignment  $\mathcal{A}$  may be written as

$$\begin{aligned} T_{\mathcal{A}}[\mathbf{b}|\mathbf{a}] = & P(S_0 \rightarrow S_1)T_{a_1}(b_1)P(S_1 \rightarrow S_2)T_{a_2}(b_2)P(S_2 \rightarrow D_3) \\ & P(D_3 \rightarrow D_4)P(D_4 \rightarrow D_5)P(D_5 \rightarrow D_6)P(D_6 \rightarrow S_7) \\ & T_{a_7}(b_3)P(S_7 \rightarrow I_7)\tilde{p}(b_4)P(I_7 \rightarrow I_7)\tilde{p}(b_5) \\ & P(I_7 \rightarrow I_7)\tilde{p}(b_6)P(I_7 \rightarrow S_8)T_{a_8}(b_7), \end{aligned}$$

where  $\tilde{p}(b)$  is the insertion probability of amino acid  $b$  between nodes 7 and 8. Assuming that  $P[\mathbf{b}] = \prod_i p(b_i)$ , one obtains the ratio

$$\begin{aligned} \frac{T_{\mathcal{A}}[\mathbf{b}|\mathbf{a}]}{P[\mathbf{b}]} = & P(S_0 \rightarrow S_1)\frac{T_{a_1}(b_1)}{p(b_1)}P(S_1 \rightarrow S_2)\frac{T_{a_2}(b_2)}{p(b_2)}P(S_2 \rightarrow D_3) \\ & P(D_3 \rightarrow D_4)P(D_4 \rightarrow D_5)P(D_5 \rightarrow D_6)P(D_6 \rightarrow S_7) \\ & \frac{T_{a_7}(b_3)}{p(b_3)}P(S_7 \rightarrow I_7)\frac{\tilde{p}(b_4)}{p(b_4)}P(I_7 \rightarrow I_7)\frac{\tilde{p}(b_5)}{p(b_5)} \\ & P(I_7 \rightarrow I_7)\frac{\tilde{p}(b_6)}{p(b_6)}P(I_7 \rightarrow S_8)\frac{T_{a_8}(b_7)}{p(b_7)}. \end{aligned} \quad (11)$$

Comparing (10) and (11) and events of similar type yields the following mappings:

$$\begin{aligned} \exp[s_i(b_j)] &= P(S_{i-1} \rightarrow S_i)\frac{T_{a_i}(b_j)}{p(b_j)} \\ \exp[g_{(3-6)}^D] &= P(D_3 \rightarrow D_4)P(D_4 \rightarrow D_5)P(D_5 \rightarrow D_6) \\ &\quad \times \frac{P(S_2 \rightarrow D_3)P(D_6 \rightarrow S_7)}{P(S_6 \rightarrow S_7)} \\ \exp[g_7^I([b_4, b_5, b_6])] &= \frac{P(S_7 \rightarrow I_7)P(I_7 \rightarrow S_8)}{P(S_7 \rightarrow S_8)} [P(I_7 \rightarrow I_7)]^2 \\ &\quad \times \frac{\tilde{p}(b_4)\tilde{p}(b_5)\tilde{p}(b_6)}{p(b_4)p(b_5)p(b_6)}. \end{aligned} \quad (12)$$

Frequently HMMs take  $P(D_{i-1} \rightarrow D_i)$  and  $P(I_j \rightarrow I_j)$  each to be a constant. In this case, the ratio  $\frac{P(S_7 \rightarrow I_7)P(I_7 \rightarrow S_8)}{P(S_7 \rightarrow S_8)P(I \rightarrow I)}$  contributes to a position specific gap opening cost and the ratio  $\frac{\tilde{p}(b)}{p(b)}$  contributes to a composition-dependent insertion cost. The quantity  $P(I \rightarrow I)$  contributes to the insertion gap extension



cost. If one keeps emission probabilities  $T_{a_i}(b)$  node-dependent, but demands that all the state-to-state transition probabilities be node-independent, one essentially has a PSSM with uniform affine gap costs, although possibly with composition-specific insertion gap costs if  $\tilde{p}$  is chosen to be different from the background  $p$ .

Since the transition probabilities are constrained by the respective conservation conditions, and those probabilities are related to the scoring function through (12), the substitution and gap scores are no longer independent if one wishes to have a probabilistic interpretation. We now turn to the relationship among score parameters when the state to state transition probabilities are node-independent constants. Let  $\eta \equiv P(S \rightarrow S)$ ,  $\mu^{D1} \equiv P(D \rightarrow S)/P(S \rightarrow S)$ ,  $\mu^{D2} \equiv P(S \rightarrow D)$ ,  $\mu^{I1} \equiv P(I \rightarrow S)/P(S \rightarrow S)$ ,  $\mu^{I2} \equiv P(S \rightarrow I)$ ,  $\nu^I \equiv P(I \rightarrow I)$ , and  $\nu^D \equiv P(D \rightarrow D)$ . Because  $\mu^{I1}(\mu^{D1})$  and  $\mu^{I2}(\mu^{D2})$  always appear together as a product, we further define  $\mu^I \equiv \mu^{I1}\mu^{I2}$  ( $\mu^D \equiv \mu^{D1}\mu^{D2}$ ). The probability conservation condition then demands that

$$\eta + \mu^{I2} + \mu^{D2} = 1 \quad (13)$$

$$\eta\mu^{I1} + \nu^I = 1 \quad (14)$$

$$\eta\mu^{D1} + \nu^D = 1. \quad (15)$$

Treating  $\nu^I$ ,  $\nu^D$ ,  $\mu^I$  and  $\mu^D$  as fixed parameters allows us to express  $\eta$ ,  $\mu^{I2}$ , and  $\mu^{D2}$  in terms of  $\nu$ s and  $\mu^{I(D)}$ . To do so, we multiply (14) by  $\mu^{I2}$  and multiply (15) by  $\mu^{D2}$ . Together with (13), we have three linear equations with three unknowns:  $\eta$ ,  $\mu^{I2}$ , and  $\mu^{D2}$ . Solving these equations yields

$$\begin{aligned} \eta &= \frac{1}{1 + \mu^I/(1 - \nu^I) + \mu^D/(1 - \nu^D)} \\ \mu^{I2} &= \frac{\mu^I/(1 - \nu^I)}{1 + \mu^I/(1 - \nu^I) + \mu^D/(1 - \nu^D)} \\ \mu^{D2} &= \frac{\mu^D/(1 - \nu^D)}{1 + \mu^I/(1 - \nu^I) + \mu^D/(1 - \nu^D)}. \end{aligned}$$

For the case  $\mu^D = \mu^I = \mu$  and  $\nu^D = \nu^I = \nu$ , these expressions simplify to

$$\begin{aligned} \eta &= \frac{1 - \nu}{1 + 2\mu - \nu} \\ \mu^{D2} = \mu^{I2} &= \frac{\mu}{1 + 2\mu - \nu}. \end{aligned}$$

Note that with this notation, we may rewrite (12) as

$$\begin{aligned} \exp[s_i(b_j)] &= \eta \frac{T_{a_i}(b_j)}{p(b_j)} \\ \exp[g_{3-6}^D] &= \mu\nu^3 \\ \exp[g_7^I([b_4, b_5, b_6])] &= \mu\nu^2 \times \frac{\tilde{p}(b_4)}{p(b_4)} \frac{\tilde{p}(b_5)}{p(b_5)} \frac{\tilde{p}(b_6)}{p(b_6)}. \end{aligned}$$

It becomes evident that  $\ln(\mu/\nu)$  corresponds to the gap opening score while  $\ln(\nu)$  corresponds to the gap extension score, and  $\ln(\tilde{p}(b)/p(b))$  becomes an additional composition-specific insertion score.

In HMMER, the local alignment is terminated by going into the end state, and the end state can be reached only from substitution states. In this context, the probability conservation equations (14) and (15) remain unchanged. However, we may allow a node-specific termination probability from the  $S$  state. This requires the introduction of a position index for the other transition probabilities. Let  $\eta_m \equiv P(S_m \rightarrow S_{m+1})$ ,  $e_m \equiv P(S_m \rightarrow E)$ ,  $\mu_m^{D1} \equiv P(D_m \rightarrow S_{m+1})/P(S_m \rightarrow S_{m+1})$ ,  $\mu_m^{D2} \equiv P(S_m \rightarrow D_{m+1})$ ,  $\mu_m^{I1} \equiv P(I_m \rightarrow S_{m+1})/P(S_m \rightarrow S_{m+1})$ ,  $\mu_m^{I2} \equiv P(S_m \rightarrow I_m)$ . However, note that  $P(I_m \rightarrow S_{m+1})$  should remain the same, because there is no direct transition  $I_m \rightarrow E$ . Thus, we may still keep both  $\mu_m^{D1} \mu_m^{D2} = \mu_m^{I1} \mu_m^{I2} = \mu$  and  $\nu_m^D = \nu_m^I = \nu$  as constants. The probability conservation condition then yields

$$\eta_m + \mu_m^{I2} + \mu_m^{D2} + e_m = 1 \quad (16)$$

$$\eta_m \mu_m^{I1} + \nu = 1 \quad (17)$$

$$\eta_m \mu_m^{D1} + \nu = 1, \quad (18)$$

the solution of which is

$$\begin{aligned} \eta_m &= \frac{(1 - e_m)(1 - \nu)}{1 + 2\mu - \nu} \\ \mu_m^{D2} = \mu_m^{I2} &= \frac{\mu(1 - e_m)}{1 + 2\mu - \nu}. \end{aligned}$$

Although  $\mu_m^{D2}$  and  $\mu_m^{I2}$  are decreased,  $\mu_m^{D2} \mu_m^{D1}$  and  $\mu_m^{I2} \mu_m^{I1}$  are kept the same as before. As a consequence, the only change is that the substitution score at each node is reduced by a node-specific constant  $\ln[1/(1 - e_m)]$  when it is not preceded by a gap state. If an alignment has deletion at node  $m$  followed by  $k$  more substitutions from node  $m + 1$  to node  $m + k$ , then the substitution score reduction starts only at node  $m + 2$  and persists to node  $m + k$ .